Image and Vision Computing 62 (2017) 28–38

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Marker-based human pose tracking using adaptive annealed particle swarm optimization with search space partitioning☆

Ashraf Sharifi, Ahad Harati, Abedin Vahedian*

Department of Computer Engineering, Ferdowsi University of Mashhad, Mashhad, Iran

ARTICLE INFO

Article history:
Received 21 April 2016
Received in revised form 1 January 2017
Accepted 22 March 2017
Available online 31 March 2017

Keywords:
Pose tracking
Marker-based human pose estimation
Particle swarm optimization
Search space partitioning

ABSTRACT

Pose estimation and tracking of an articulated structure based on data from multiple cameras has seen numerous applications in recent years. In this paper, a marker-based human pose tracking algorithm from multi view video sequences is proposed. The purpose of the proposed algorithm is to present a low cost motion capture system that can be used as an alternative to high cost available commercial human motion capture systems. The problem is defined as the optimization of 45 parameters which define body pose model and is solved using a modified version of particle swarm optimization (PSO) algorithm. The objective of this optimization is to maximize a fitness function which formulates how much the body model matches with *2D* marker coordinates in video frames. A sampling covariance matrix is used in the first part of the velocity equation of PSO and is annealed with iterations. The sampling covariance matrix is computed adaptively, based on variance of parameters in the swarm. One of the concerns in this algorithm is the high number of parameters to define the model of body pose. To tackle this problem, we partition the optimization state space into six stages that exploit the hierarchical structure of the skeletal model. The first stage optimizes the six parameters that define the global orientation and position of the body. Other stages relate to optimization of right and left hand, right and left leg and head orientation. In the proposed partitioning method previously optimized parameters are allowed some variation in each step that is called soft partitioning. Experimental results on Pose Estimation and Action Recognition (PEAR) database indicate that the proposed algorithm achieves lower estimation error in tracking human motion compared with Annealed Particle Filter (APF) and Parametric Annealing (PA) methods.

1. Introduction

Capturing and tracking the human motion have found numerous applications in recent years. Motion capture is the process of recording human motion as a sequence of 3D Cartesian coordinates called motion data [1]. Human pose tracking is the process of determining the configuration (orientation and location) of body parts at consecutive time instants using motion data. There are three major goals for human pose tracking [2]: smart surveillance, object control and research purposes. The purpose of surveillance applications is human body pose tracking while monitoring for specific actions such as shop lifting. Animating virtual characters in games and movies can be considered as control applications. The aim of these applications is avatar control within virtual worlds based on human motion in the real world. Motion data in research applications are used for diagnostics in orthopedic patients in clinical studies or train athletes to improve their performance.

General structure of motion analyzing systems consists of four steps, namely initialization, tracking, body pose estimation and action recognition. Determining an appropriate model of subject in model-based systems and camera calibration in image-based systems are examples of initialization. The aim of the tracking step is to determine the position of corresponding segments of the body parts in successive frames. In the pose estimation step relative orientation and location of body parts related to each other is determined. In the last step of this process the estimated pose in consecutive frames is analyzed to recognize the action performed by the subject. There are different sensor types to capture human motion, which are categorized as active and passive sensors [3]. Active sensors transmit or receive signals from the other sensors while passive sensors have no effects on the other sensors. Accelerometers, mechanical, electromagnetic [4] and acoustic sensors are examples of active sensors

☆ This paper has been recommended for acceptance by Robert Walecki.
* Corresponding author.
E-mail addresses: sharifi.a7@alumni.um.ac.ir (A. Sharifi), a.harati@um.ac.ir (A. Harati), vahedian@um.ac.ir (A. Vahedian).

http://dx.doi.org/10.1016/j.imavis.2017.03.003
0262-8856/© 2017 Elsevier B.V. All rights reserved.

already used for human motion capture. The methods based on these sensors usually require devices to be attached to the body parts such as skeletal-like structures in mechanical approaches and magnetic or acoustic sensors in other approaches [5]. The major problem about the above methods is that the subject need to wear special suit which prevents free movement. In the methods based on passive sensors which are mainly image-based, a passive device (camera) is used to capture human motion. The objective of image-based pose estimation methods is estimating body pose which is most consistent with the image information. Image-based methods are also classified into two categories, namely marker-based and marker-less methods. Marker-less methods take video frames as input data, whereas marker-based methods rely on a number of markers placed on different parts of the human body while the scene is simultaneously captured by a number of calibrated cameras [6]. The major difficulty about marker-less methods is due to *3D* to *2D* projection and the huge amount of information contained in an image. Therefore marker-based methods are presented, to alleviate this problem [7]. This paper focuses on marker-based human pose tracking in multi view scenario to present a low cost multi camera algorithm for human motion. The problem with high cost commercial human motion capture systems is that they usually need sophisticated software and expensive hardware to work with a high number of markers which make use of these systems not affordable for small research labs. Therefore, the proposed algorithm tries to reproduce similar results with much cheaper setups using only a fraction of available input data. This algorithm relies on optimization methods for body pose tracking and uses *2D* coordinates of markers in camera views as the input parameter. Since the likelihood function in human motion tracking can get very complex form with multiple local maxima, we suggest a Mont-Carlo-based stochastic optimization algorithm for human pose estimation.

2. Related work

The objective of pose estimation algorithms is estimating the configuration of underlying skeletal structure of human body. For this purpose some algorithms use a predefined model which represents kinematic structure of human body, while others do not utilize an explicit model of human body. According to this, pose estimation algorithms are divided into model-based and model-free methods. Some of these algorithms first find possible positions of different body parts in images and then estimate the most probable body configuration based on these locations while others directly employ a classifier to find the best matched pose in database with images [8]. The majority of image-based pose estimation methods use a kinematic structure, with specified joints and degree of freedom, as the prior model. Most model-based algorithms rely on optimization methods for body pose tracking. These methods consist of parametric, non-parametric [9] and local optimization algorithms [10]. Given an initial state, local optimization moves with local changes among possible states in the space of solutions until an acceptable state is found. Performance of these methods is strongly dependent on the initial state and one of the major problems of them is local optima convergence. In local optimization methods, only one hypothesis propagates over time. Likelihood function in human pose estimation problem has multiple peaks and local maxima. Therefore, methods that rely on one hypothesis may not be able to find all these local maxima. Another major drawback of these methods is error accumulation. For example, estimated pose may be far from correct body pose due to complex body motion or imperfect observation in some frames. Correct body pose in next frames cannot be always recovered in such situations [9]. Stochastic optimization techniques are then introduced, to overcome these problems. These methods are based on multiple hypothesis propagation that would make the tracking process more robust [11]. In human pose estimation each hypothesis represent a possible pose of human body. Genetic Algorithm (GA) is an example of stochastic methods that is based on a population of candidate solutions. It starts with randomly generated individuals. Fitness of every individual is evaluated and best ones are selected in an iterative process. Then a combination of crossover and mutation operations are performed on these individuals and the next population are generated. In [12] GA algorithm was proposed to estimate upper-body pose and Zhao and Liu presented an Annealed Genetic Algorithm (AGA) for *3D* human motion analysis [13]. Particle Filter (PF) or Condensation algorithm [14] is one of the useful methods for pose tracking purpose. In this method posterior probability is approximated using a set of weighted particles. In each iteration the likelihood of particles is computed followed by a resampling operation to remove particles with low weights and concentrate particles to more likely pose. In [15] particle filter is used to pose estimation of human body. Two common problems about particle filter are sample impoverishment and needs for high number of particles in high dimensional state space. In resampling operation, particles are selected with probability proportional to their weights to remove particles with low weights and concentrate particles to more likely state. Therefore, particles with large weights are likely to be selected multiple times, whereas the other particles with small weights, are not likely to be drawn at all. This causes a problem called “sample impoverishment” in which the number of distinct samples are reduced and negatively impacts on distribution representation. One way to solve this problem is enlarging the sample set to cover state space completely, which increases the computational load. Another challenge about particle filter is that for successful tracking, the required particles’ count increases exponentially with the dimension of the state space. More than 20 parameters are required to describe a realistic articulated model of human body. As a result, particle filter algorithm needs huge number of particles to approximate the underlying probability distribution in the body space. One way to alleviate this problem is to use simulated annealing idea in particle filter algorithm. In 2000, Deutscher [16] introduced a modified particle filter, named Annealed Particle Filter (APF), that required fewer particles and could get better results compared with standard particle filter. To better explore the search space, instead of using single weighting function in PF, APF uses a series of weighting functions (W_0 to W_M) in which each W_m differs slightly from W_{m-1}.The first function W_m is designed so smooth to represent the overall trend of search space while W_0 emphasizes its local features. Thus the initial searching area is global at first and gradually becomes local within layers. In 2005, an improved version of APF was proposed that used crossover operation and search space partitioning [17]. This version of APF was used as the baseline algorithm in Human Eva framework [18]. APF is one of the most used algorithms in pose estimation area [7]. Pose estimation algorithms presented in [16] and [19] used APF algorithm to marker-less human pose estimation. In [7] and [20], APF is used for marker-based pose estimation and pose estimation-based on *3D* point cloud of human body, respectively. As mentioned in [21], probability distribution in particle filter-based methods can not be explored efficiently since particles have no relationship to each other and do not move according to their former experience, which reduce capability of samples to escape local minima. This paper also reports that although the performance of annealed particle filter in images with frame rate of 60 fps is acceptable, but the performance is reduced in frame rate below 30. Particle Swarm Optimization(PSO) is another useful method for body pose tracking.PSO is an evolutionary optimization algorithm that was used in 2006 to solve upper-body pose estimation [22] and has recently attracted more attention for full-body pose estimation [11–14]. The communications of particles in PSO has led to more efficient search than particle filter-based methods and the crossover operation in GA [17]. One of the major challenges in this problem is the high number of DOF (degree of freedom) that has to be recovered. An effective method to reduce search complexity in such problems is the search space partitioning. In this method one section of search space is optimized independently and its result is used as a constraint to limit the rest of the search space [23]. The objective of all optimization methods is to determine model state which is most consistent with observations. Therefore, these methods often use a likelihood function to determine how well a body pose fit to current observation. Likelihood computation in most of marker-less pose tracking algorithms is based on edge, silhouette or combination of them in images [24]. In these algorithms, likelihood of body pose is computed based on the amount of overlap between observed silhouette and projected pose on image [25]. The objective of method introduced in [20] is to estimate pose of human body in *3D* point clouds recorded with a *3D* sensor. In this method each body part is modeled using a truncated cylinder and likelihood is computed based on matching of skeleton points and cylinder edges against the point cloud, and reverse-matching the point cloud against the skeleton points. In this paper a modified version of PSO algorithm for marker-based human pose tracking is presented. in order to reduce search complexity in such a problem we use search space partitioning. We have presented the initial version of this algorithm in [26] previously. In this paper, in addition to providing further experiments compared with [26] an improved version of algorithm is presented. In the proposed algorithm, we adaptively compute velocity of particles based on variance of the parameters that leads to better tracking result. This method is discussed in Section 3.4 in detail. In Section 4.9 we compare the proposed algorithm with the initial version of algorithm, presented in [26]. As can be seen in the results section (Section 4), the proposed algorithm can achieve good performance for human pose tracking in low frame rate (25 fps) regardless of the complexity of the selected human body model (41 parameters).

3. Proposed algorithm

As mentioned before, Bayesian tracking formulation is one of the major methods to represent pose tracking. The objective of this formulation is to predict posterior probability distribution ($p(x_{t-1} \mid y_{1:t-1})$).Where x_t is current state, in this application current body pose whose motion is to be captured, and $y_{1:t}$ is the observation up to current time. Based on Bayesian formulation posterior probability distribution is represented as a hidden Markov model:

$$p(x_t \mid y_{1:t}) \propto p(y_t \mid x_t) \int p(x_t \mid x_{t-1}) p(x_{t-1} \mid y_{1:t-1}) dx_{t-1} \tag{1}$$

This recursive formula is based on Markov assumption that current pose of body at time t depends only on previous pose at time $t-1$ (Eq. (2)) and current observation only depends on the current body pose (Eq. (3)):

$$p(x_t \mid x_{1:t-1}) = p(x_t \mid x_{t-1}) \tag{2}$$

$$p(y_t \mid x_{1:t}, y_{1:t-1}) = p(y_t \mid x_t) \tag{3}$$

Using this formula, human body pose tracking is explained as a two-step process that consists of prediction and update steps. Previous pose of body is used to predict current pose in prediction step. Depending on how well this prediction fit to current image information, likelihood is evaluated in update step. The proposed method for these steps is described as below.

3.1. Human body model

As mentioned before, model-based algorithms use a predefined model which represents kinematic structure of human body. The kinematic structure of human body is usually represented as a *3D* kinematic tree in which each node corresponds to a joint in the human body. Torso joint of the human body is usually assumed as the root node in kinematic tree. Every node has up to three rotational DOF, while the root node also has three translational DOF that determines the global position of the body. Joint positions are centers of rotation in each part's coordinate frame. Relative position and orientation of each part is represented using a local homogeneous transformation matrix:

$$T = \begin{pmatrix} R_x R_y R_z & t \\ 0 & 1 \end{pmatrix} \tag{4}$$

where R_x, R_y, and R_z are local rotation matrices with three Euler angles about x, y, and z coordinate axes and t is translation vector to specify relative position of a joint and its parent. So each part in body model has a local transformation matrix. Kinematic tree specifies the order of transformations between body parts. To create global transformation matrix related to the root node of the tree, we multiply transformation matrix of each part to its parent matrix and continue until we reach the root of the hierarchy [27].

$$T_g = T_{child} * T_{parent} * T_{grandparent} \cdots * T_{root} \tag{5}$$

Each point in local coordinate system can be transformed to global coordinate system, using global transformation matrix of each joint. In the tracking process joint angles are variable parameters during the tracking while length of body limbs is assumed to be constant. In this paper, we use 45 parameters including global position and joint angles to represent the full body model.

3.2. Propagation model

Pose tracking process is performed by maximizing a fitness function in image sequences. Population in each frame is generated from the estimated pose in previous frame. A motion model is employed to propagate particles to the new frame, to initialize optimization in each frame. Zero motion with additional gaussian noise is almost used as motion model in tracking algorithms [17,20,24,28].

$$x_i^t \leftarrow N(\hat{x}^{t-1}, \Sigma), \Sigma = \begin{pmatrix} \sigma_1^2 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_{45}^2 \end{pmatrix}, \sigma = \begin{pmatrix} \sigma_1 \\ \vdots \\ \sigma_{45} \end{pmatrix} \tag{6}$$

where standard deviation σ_d in the Σ matrix is equal to the maximum absolute inter-frame differences of the joint angles that are almost determined in a training process. Experiments on the motion model conducted in [18] show that tracking accuracy largely depends on the amount of standard deviation values in the Σ matrix. These experiments show that the most accurate results were obtained when style-specific motion model was used. For example to track walking motion only data related to walking motion is used to learn the sampling covariance matrix (Σ). In the proposed algorithm, we use a generic motion model in which the amount of standard deviation for all joints and for all motions is set to a low value (0.1 radian in this paper). This motion model eliminated the need for training process. As we show in the Experimental results section, our algorithm can track different motions with no prior knowledge of motions type using this motion model.

3.3. Likelihood evaluation

In order to evaluate likelihood of a body pose based on observation data, we use the method presented in [7] which is explained in this section. Likelihood computation in this method is performed based on generalized epipolar geometry in four cameras [29]. When two pinhole cameras see a *3D* scene from two different positions, a geometric relation called epipolar geometry is occurred between projections of each *3D* points onto *2D* images. Based on the epipolar geometry, correspondence of a *2D* point in an image is located on specific line in the second image that is called epipolar line. Epipolar geometry can be extended for more cameras and more views. If $l(x^i, j)$ be the epipolar line generated by the point x in a given view i onto another view j and $d(l(x^i, j), x^j)$ is defined as Euclidean distance between the epipolar line $l(x^i, j)$ and the point x^j, symmetric epipolar distance in two views can be defined as (Eq. (7)):

$$d_{s\epsilon}(p^i, p^j) \triangleq \sqrt{d^2(l(x^i, j), x^j) + d^2(l(x^j, i), x^i)} \tag{7}$$

Based on this definition, extension of the symmetric epipolar distance for $k \geq 2$ points in k different views ($d_{s\epsilon}(x^1, \dots, x^k)$) can be computed as (Eq. (8)):

$$d_{s\epsilon}(x^1, \dots, x^k) = \sqrt{\sum_{i=1}^{k-2} \sum_{j=i+1}^{k-1} d^2_{s\epsilon}(x^i, x^j)} \tag{8}$$

where $d_{s\epsilon}(x^j, x^j)$ is symmetric epipolar distance between two points x^i and x^j in the two views i and j. Using these formulas likelihood is computed as follows. The *2D* coordinate of markers attached to the body of the performer onto *NC* camera views (four views in this paper) provide observation data (z_t). If $D_n = d_1, d_2, \dots, d_{Q_n}$ is the set of Q_n, *2D* coordinate of markers in the n-th view and $X = \{x_1, \dots, x_M\} \in R^3$ is the *3D* position of the body joints and the end of the limbs using forward kinematics. The computed fitness function should measure how well these *2D* coordinates fit as projections of *3D* coordinate of the set X. For every element x_m in the set X, it's projection onto every camera view is computed in (Eq. (9))

$$P_{m,n} = P_n(x_m), 1 \leq m \leq M, 1 \leq n \leq N_c \tag{9}$$

where P_n is the perspective projection operator from *3D* to *2D* on the nth view. Next the set $T_m = t_1, \dots, t_{NC}$ containing the closest measurement in every camera view associated to every element x_m is constructed according to (Eq. (10)):

$$t_n = min_{d_q} \parallel p_{m,n} - d_q \parallel, d_q \in D_n, \forall n \tag{10}$$

Not all *3D* points may have projection on images because of occlusion. To detect such cases, we applied a thresholding with a value equal to 5 pixels ($\rho = 5$) on elements of t_n. So if $\parallel p_{m,n} - d_q \parallel > \rho, t_n = \emptyset$ is considered. In these cases we use a penalty value equal to 200 as symmetric epipolar distance. Then the extension of the symmetric epipolar distance for $k \geq 2$ points in k different views (Eq. (8)) is used to compute fitness value. When the *2D* points are projected from the same *3D* point, this distance decreases. With this definition, the score s_m and weighting function W over the M position of body joints is formulated in (Eqs. (11)–(12)):

$$s_m(z_t, x_m) \equiv s_m(z_t, T_m) \propto d_{s\epsilon}(T_m) \tag{11}$$

$$w(z_t, y) = \exp\left(-\frac{1}{M} \sum_{m=1}^{M} (s_m(z_t, x_m)\right) \tag{12}$$

3.4. Optimization method

We use a modified version of PSO algorithm for full body pose estimation. As mentioned before, PSO is a computational method that is used to solve optimization problems. This algorithm is based on a population of candidate solutions. One of the most important properties of PSO is that unlike the other particle-based methods, such as particle filter and its variants, particles share their information with each other and with the best particle in the whole population. Therefore the search is more efficient than the crossover operation in GA. PSO is based on a swarm consisting of N particles. Let x^i be the i-th particle, v^i be the velocity of it, p^i be the best position of i-th particle that encountered so far and p^g be the best known position of the entire swarm. With these assumptions, the classic PSO algorithm can be explained as follows:

- Randomly initialize the population's position and velocity in the search space.
- Set p^i for each particle and identify best particle in the swarm and set as p^g.
- Repeat the two following steps until stopping criterion is satisfied:

    1. Compute velocity of particles according to Eq. (13) and update the position of every particle x^i based on Eq. (14):

    $$v^i_{t+1} = v^i_t + \varphi_1(p^i_t - x^i_t) + \varphi_2(p^g_t - x^i_t) \tag{13}$$

    $$x^i_{t+1} = x^i_t + v^i_{t+1} \tag{14}$$

    2. Update p^i, p^g for each particle

where subscript t denotes the time step (iteration). Parameters φ_1 and φ_2 are random numbers drawn from [0;1]. Criterion for termination is usually the maximum number of iterations.

Velocity equation of PSO algorithm (Eq. (13)) consists of three components [30].

1. The first component is known as "habit" or "history" and makes particles continue moving in the same direction they have been traveling so far.
2. The second component is called memory or self-knowledge which attracts particles to the best position ever reached by each particle.
3. Third part of this equation is known as "shared information" or "social knowledge" and attracts particles to the best position that ever found by all particles.

The history component in classical PSO is initialized randomly. Due to the complex motion of body parts, history of particles may not be much reliable. So random initialization of this part leads to produce many impossible body poses. To solve this problem, we use the method presented in [31]. In this method, the sampling covariance matrix is used to initialize the first part of the equation (Eq. (13)) that annealed during iterations. Thus the initial searching area is global at first and gradually becomes local during the iterations. The velocity equation (Eq. (13)) is updated as (Eq. (15)):

$$v^{i,n+1} = P_n + \varphi_1(p^i - x^{i,n}) + \varphi_2(g - x^{i,n}) \tag{15}$$

Given the sampling factor $\alpha_n < 1$, the covariance matrix P_n is evolved as follows:

$$P_n = \alpha_n * P_0 \tag{16}$$

where P_0 is the covariance matrix described in Propagation model section. In the proposed algorithm, sampling factor α_n is formulated in (Eq. (17))

$$\alpha_n = -0.8 * \left(\frac{n}{M}\right) + 1 \tag{17}$$

In this equation n is the current number and M is the maximal number of iterations. As can be seen in Eq. (16), for all parameters of body model and in each iteration, P_n matrix is computed similarly using the same coefficient (α). Variance of some parameters may be greatly reduced in specific iteration. As can be seen in Eq. (16), values of P_n matrix is calculated regardless of the variance of parameters. In this situation, adding P_n matrix to particles causes them to be dispersed after convergence. We have presented this version of algorithm in [26] previously.

To alleviate this problem, we determine P_n matrix proportional to the variance of the parameters in swarm (Eq. (18)). Using this idea, the tracking results can be improved especially in low frame rates.

$$P_n = \alpha_n * Var(X_{n-1}) \tag{18}$$

where X_{n-1} is the set of particles in iteration $n-1$ and var is the variance operation. One of the major challenges in this problem is the high number of DOF that has to be recovered. An effective method to reduce search complexity in such problem is the search space partitioning. In this method, one section of search space is optimized independently and its result is used as a constraint to limit the rest of search space. In this work, we propose to split the 45-dimensional search space into six stages that each one is optimized using optimization algorithm described above. In the first stage, six parameters related to global position and orientation of torso are optimized. Other five stages are related to optimization of left and right hand, left and right leg and head orientation. In [24] search space partitioning is separated into hard and soft partitioning categories. In hard partitioning each step is optimized while the other steps are kept constant. So if some errors occur in the first step, the optimizer can not correct these errors in the following steps. In this situation errors spread and accumulate during the hierarchical steps. Whereas in soft portioning, previously optimized parameters are allowed some variations in each step. In this work the standard deviations for first six optimized parameters are reduced to one tenth in the five stages and a trivial search about the result of stage one is performed in the other stages to avoid error accumulation.

4. Experimental results

To test the proposed algorithm, we use PEAR database [32]. This database contains synchronized motion capture and 16 visual streams from different views at resolution of 704 * 576 pixels and frame-rate of 25 fps. As the other papers in the field of human pose tracking, performance of the proposed algorithm is evaluated based on ground truth data prepared in database which is obtained by a commercial motion capture system from MotionAnalysis[1]. In this system thirty markers are attached onto the key joint positions of body and 12 cameras with over 1-M-pixel resolution with frame rate of 50 fps are used to estimate the *3D* articulated pose of the body. While in proposed algorithm four camera images with resolution of 704 * 576 pixels and frame rate of 25 fps are used to produce ground truth data, which is half of the frame rate of motion capture system.

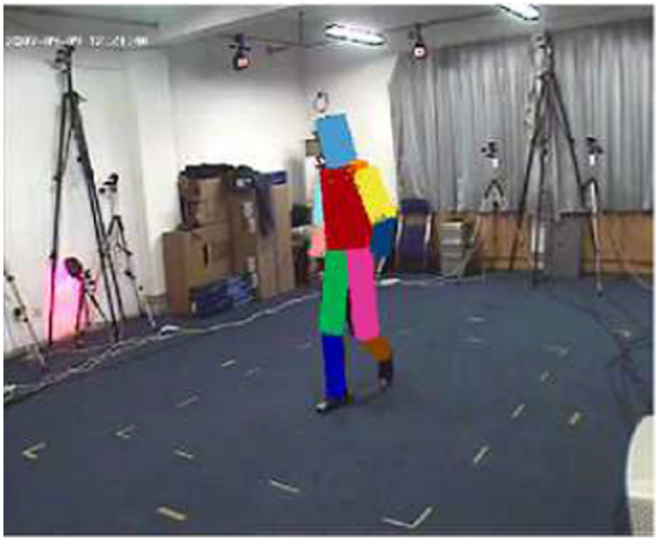


Fig. 1. Synthetic data generation — auto occlusion modeling among different body parts.

PEAR database contains 5 subjects performing 6 predefined actions. The predefined actions include walking, jumping, skipping, waving, stretching and jogging motion and ground-truth data is available. To carry out the test of performance, the metrics presented in [18] were used which included mean and standard deviation (σ) of the estimation error and two metrics (MMTA[2] and MMTP[3]) proposed in [33]. Let $\hat{x}$ be the landmark position associated to estimated pose and x be the ground truth position. With these assumptions, above evaluation metrics can be defined as follows. In these metrics, euclidean distance between the estimated landmark and ground truth is considered as estimation error. Based on this definition, mean and standard deviation of estimation error are computed as follows:

$$\mu = \frac{1}{M}\sum_{i=1}^{M} \parallel x_i - \widehat{x_i} \parallel, \sigma = \sqrt{\frac{1}{M} * \sum_{i=1}^{M}(x_i - \mu)^2} \tag{19}$$

MMTA is percentage of $\hat{x}$ positions that are closer than δ (10 cm in this paper) to the ground truth positions (Eq. (20)):

$$\epsilon = \parallel x_m - \hat{x}_m \parallel < \delta \tag{20}$$

MMTP is the average of distance between $\hat{x}_m$ and x_m for all these pairs. Finally, the average of these metrics for all frames is computed. To analyze various experiments, first the effect of algorithm parameters such as number of particles and iterations are examined. Then performance of the algorithm in low frame rate and computation time for each stage of optimization is reported. Finally, performance of the algorithm is compared with the initial version of the algorithm presented in [26], the annealed particle filter and parametric annealing algorithm.

4.1. Synthetic data generation on PEAR database

Proposed algorithm relies on fifteen *2D* marker coordinates over N_C camera views as the input data. We implement a synthetic data generation method to compute *2D* marker coordinates. In this

[1] http://www.motionanalysis.com.

[2] Multiple marker tracking accuracy.

[3] Multiple marker tracking precision.

Table 1
Result of proposed tracking algorithm at 25fps with base configuration.

Motion	MMTA (%)	MMTP (mm)	μ (mm)	σ (mm)
Stretch	99.11	7.85	8.97	4.30
Wave	99.96	6.91	6.95	1.91
Walk	99.88	5.94	6.05	3.04
Jog	99.41	6.76	6.84	2.04
Jump	93.97	14.01	21.14	16.54
Skip	97.95	15.61	18.07	6.27
Mean	98.38	9.51	11.33	5.68

method, *3D* ground truth data are used to compute *2D* projection of the markers onto all camera views that is presented as follows. first we apply inverse kinematic to ground truth data in order to compute correct body pose in each frame. Then *3D* coordinate of body joints (X_t) are projected onto every camera view in order to generate the sets $D_n, 1 \leq n \leq N_C$.

We use following method to model auto occlusions among body parts to check visibility of markers in each camera view. In this method each body part is modeled as a cylindrical mesh with specific color. Given the internal parameters of each N_C cameras, depth of each *3D* point of these cylindrical meshes for each camera view can be computed. In the next step, we sort *3D* points of cylindrical mesh by their ascending depth values in that camera in order to generate set $S_n, 1 \leq n \leq N_C$, for each camera view n. Then we project *3D* locations of S_n set respectively onto view n (Fig. 1). As can be seen in this figure, auto occlusion among different body parts and therefore visibility of markers can be determined by color analyzing. Some of this marker positions are removed to simulate real marker detection algorithm. Finally, a number of false detections are generated and the amount of Gaussian noise is added to some positions randomly.

4.2. Experimental setup

The base configuration of the proposed algorithm for six stages is defined as follows:

- Stages 1–5: 80 particles, 10 iterations
- Stage 6: 30 particles, 10 iterations

This setup is used for all experiments in this section unless otherwise specified. Stages 1 to 6 are related to torso position and orientation, left and right hand, left and right leg and head orientation. Since the proposed method is a sample of stochastic algorithms, tracking results slightly differs in each run. Therefore, each experiment is carried out 5 times and mean of errors is reported to test repetitively of the proposed algorithm. In some frames, none of the estimated landmark positions are closer than 10 cm to ground truth positions (σ in Eq. (20)). In these situations, we use a penalty value equal to 10 cm as distance of landmark positions to ground truth to compute MMTA metric.

4.3. Results in base configuration

Table 1 and Figs. 2 and 3 show the results of proposed algorithm with base configuration. Result of this table indicates that in 98.38% of evaluated frames, difference between ground truth and estimated pose is below 10 cm and mean of the error is 11.33 mm. The little

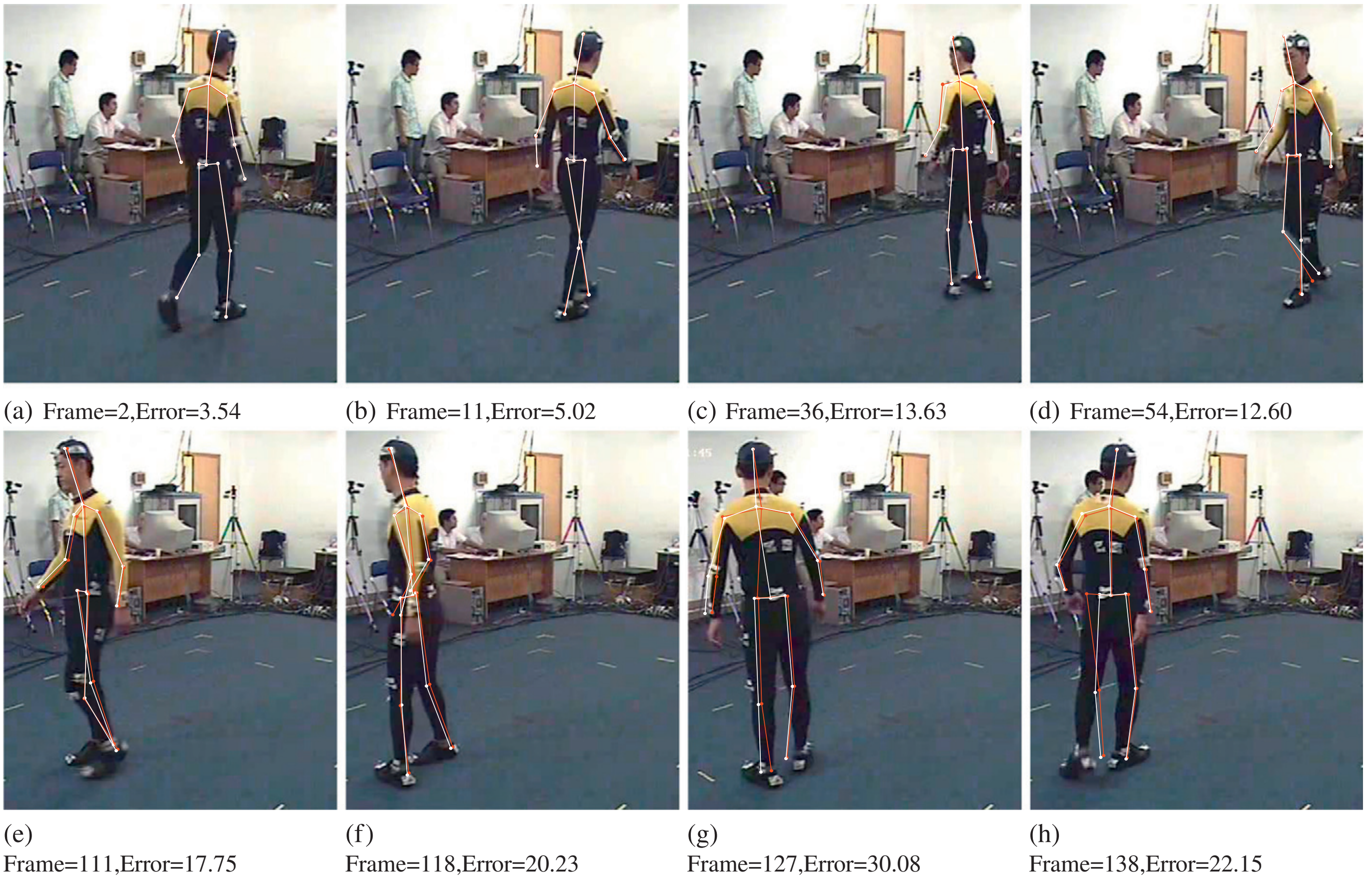
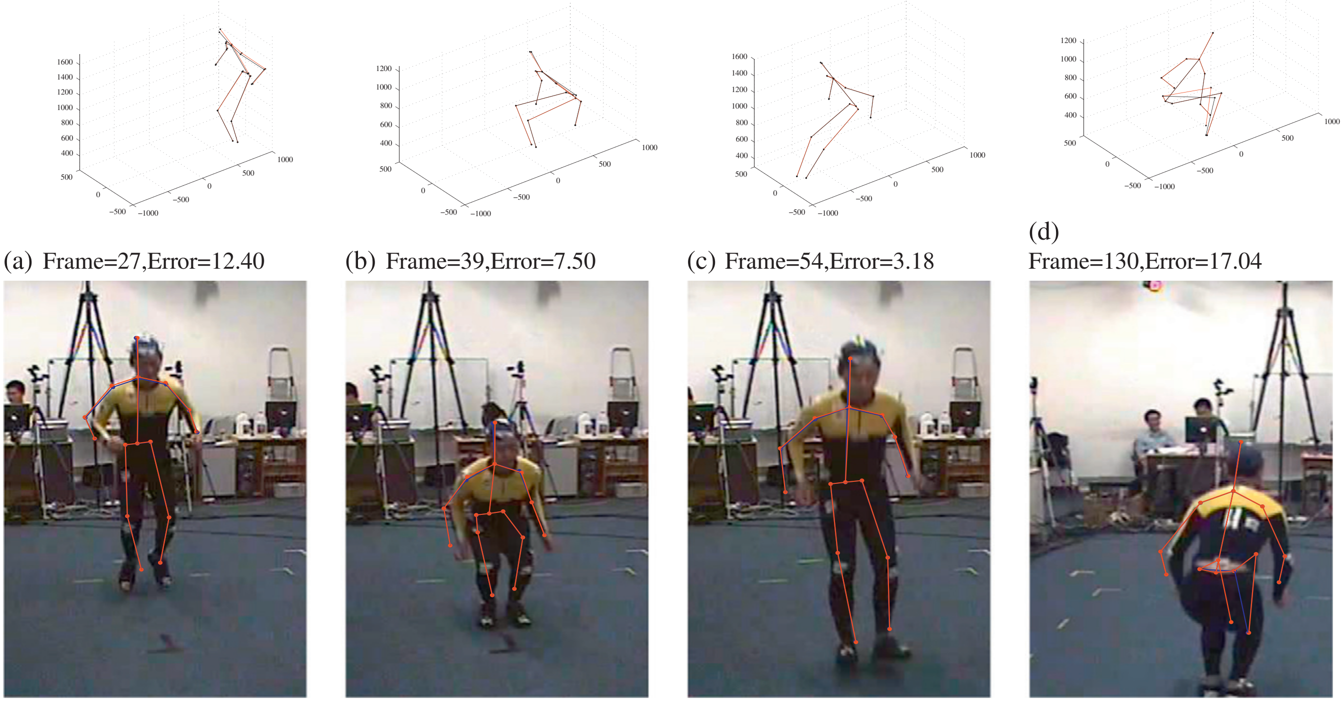

(a) Frame=2,Error=3.54
(b) Frame=11,Error=5.02
(c) Frame=36,Error=13.63
(d) Frame=54,Error=12.60
(e) Frame=111,Error=17.75
(f) Frame=118,Error=20.23
(g) Frame=127,Error=30.08
(h) Frame=138,Error=22.15

Fig. 2. Mean tracking error (mm) for sample frames from walking motion: projection of the ground truth skeleton onto the image is shown in white and projection of estimated skeleton is colored red at depicted frame. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) Frame=27,Error=12.40 (b) Frame=39,Error=7.50 (c) Frame=54,Error=3.18 (d) Frame=130,Error=17.04

(e) Projection of (a) (f) Projection of (b) (g) Projection of (c) (h) Projection of (d)

Fig. 3. Tracking result for jump motion, E denotes mean tracking error (mm) at depicted frame (first row) 3D result: ground truth skeleton is shown in black and estimated skeleton is colored red (second row) projection of ground truth skeleton on image is shown in red and estimated skeleton is colored blue. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

value of σ (5.68) means that estimation errors in total experiments are closer together which reflects the stability of the proposed algorithm. As can be seen in this table, the algorithm is less efficient for complex motions such as jump and skip, compared with motions such as wave and walk. The reason is these motions have more challenges like complex motion of hands and legs and change the direction of subject.

Error graphs of proposed approach for stretch and skip motions are shown in Fig. 4. Error graph is a graphical representation of mean tracking error that represent mean error values during a specific action tracking and is more informative than total error reported in Table 1. Maximum error in skip motion on this error graph is related to frames which contain direction change of subject and maximum error for stretch motion is occurred in frames that hand motion of subject is more complex. As mentioned in the previous sections, the proposed method is based on the soft partitioning of the search space. In this method the previously optimized parameters are allowed some variation in each optimization step. Therefore, if some errors occur in each step the optimizer can correct these errors in the following steps. This prevents error accumulation during the hierarchical steps. As the graph illustrates, tracking error in a few frames after these maximum errors is reduced which indicates the ability of the proposed method to recover the correct pose after a wrong estimation.

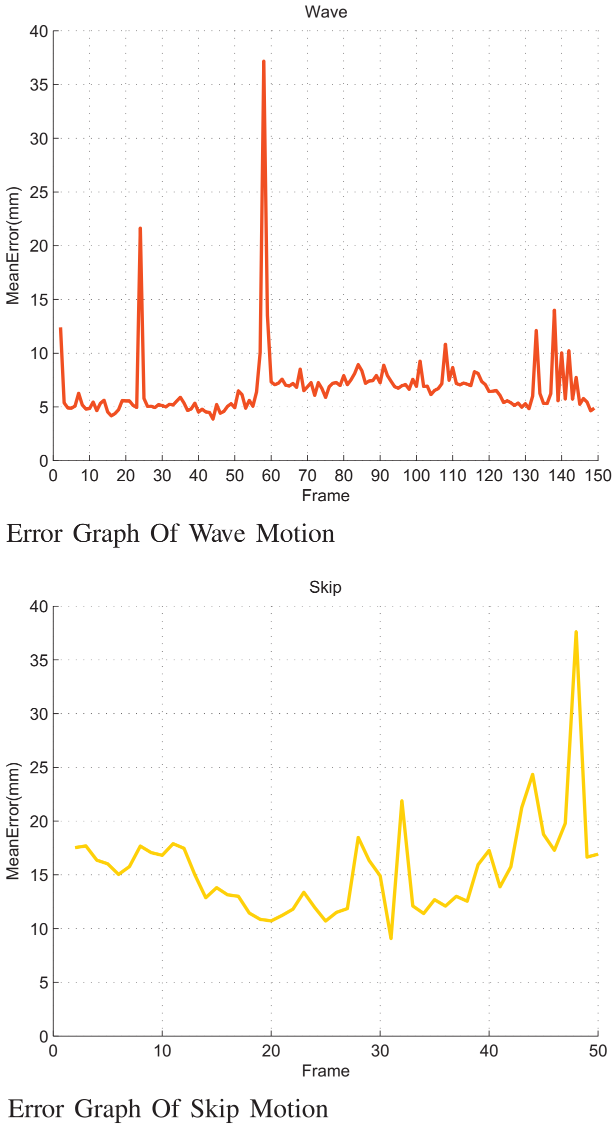
In order to test the effect of the number of particles on the performance of the pose tracking algorithm, we run the proposed algorithm with different particle sizes while keeping total number of fitness evaluations equal to the base configuration. Table 2 shows the mean result for 5 runs in different configurations. As indicated in Table 2, the performance of tracking with low particle size (*config*1) or low iteration number (*config*4) is not good enough, while the algorithm performs well in medium size of particles and iterations (base configuration and config 3).

4.4. Performance in low frame rate

In this section, the performance of the proposed algorithm is evaluated in low frame rate. The values of matrix P_0 in Eq. (16) used for 25 fps are doubled and is used as the initial covariance matrix in Eq. (15) for 12 fps. As mentioned in the previous sections we remove training process in particle propagation step between frames and we use a generic motion model for all type of motions. So proposed method is independent of type of the motion. Table 3 shows the results of the proposed algorithm with base configuration at 12 fps. According to this table, the algorithm achieves good performance in all motions except jump action.So the proposed method is independent of frame rate also, using generic propagation model. Difference of parameters in two consecutive frames is high in jumping motion. It seems that more sophisticated model is needed for particle propagation between two consecutive frames, to track this motion.

4.5. Search space partitioning

In this section we analyze the effect of search space partitioning on the performance of the algorithm. Table 4 indicates the results of the proposed method with global optimization. In this experiment, we use 400 particles and 10 iterations to perform optimization of 45 parameters in one stage. Tracking result of the proposed method with search space partitioning and global optimization is reported in Tables 1 and 4, respectively. Comparison of tracking result in these tables indicates that the performance of algorithm with search space



(a) Error Graph Of Wave Motion

(b) Error Graph Of Skip Motion

Fig. 4. Error graph of tracking result with base configuration.

Table 3
Result of the proposed algorithm with base configuration at 12fps.

Motion	MMTA (%)	MMTP (mm)	μ (mm)	σ (mm)
Stretch	95.89	8.54	24.11	14.84
Wave	99.78	6.91	16.90	7.23
Walk	90.88	8.37	29.72	23.70
Jog	87.54	11.89	35.44	36.78
Jump	47.45	49.58	567.86	319.53
Skip	92.50	17.33	28.58	30.18
Mean	85.67	17.10	117.10	72.04

partitioning is much better compared with global optimization. Tracking result in Table 4 shows that despise of using high number of particles (400 particles), global optimization can not achieve any good performance. The reason is that in this problem high number of DOF has to be recovered. So searching process is very complex, therefore we need a method to reduce search space complexity. As mentioned before we propose to split the *45-dimensional* search space into six stages. The result of previous sections shows that the proposed algorithm achieves good performance using this method, despite of small size of particle set for each stages.

4.6. Hard partitioning vs soft partitioning

In these experiments we analyze the impact of the search space partitioning method on performance of the proposed algorithm. As mentioned before, we use the soft partitioning method to reduce the search space complexity. In this method, previously optimized parameters are allowed some variations in each step of the search. Whereas in hard partitioning method each step is optimized while the other steps are kept constant. Tables 1, 4 and 5 show the result of using soft partitioning, global optimization and hard partitioning in the proposed algorithm, respectively. The results presented so far show that hard partitioning leads to the better result of the proposed algorithm compared with global optimization but the best overall results occurs when the soft partitioning method is used with the proposed algorithm. This is may be because of the ability of the soft partitioning method to prevent error accumulation by correct the errors that happened in the specific step in the following steps.

Table 2
Effect of the number of particles on the performance of proposed algorithm — result of proposed algorithm with different configurations (*p* denotes the number of particles and *i* denotes iterations).

	Config1 (stages 1–5: 10 P, 80 I–stage 6: 30 P, 10 I)				Config2 (stages 1–5: 20 P, 40 I–stage 6: 30 P, 10 I)			
	MMTA (%)	MMTP	μ	σ	MMTA (%)	MMTP	μ	σ
Stretch	92.42	9.44	315	169	95.07	9.03	17.04	13.31
Wave	90.16	8.74	395	341	96.06	7.47	13.75	10.57
Walk	51.22	22.79	2933	2454	97.18	3.68	10.42	8.47
Jog	49.65	31.56	3841	3021	94.25	5.78	17.51	15.67
Jump	31.27	56.59	4328	3479	36.26	61.99	206.44	96.13
Skip	40.27	48.58	4839	4803	82.99	22.41	49.72	52.86
Mean	59.17	29.61	2775	2378	83.64	18.39	52.48	32.83
	Config3 (stages 1–5: 40 P, 20 I–stage 6: 30 P, 10 I)				Config4 (stages 1–5: 400 P, 2 I–stage 6: 30 P, 10 I)			
	MMTA (%)	MMTP	μ	σ	MMTA (%)	MMTP	μ	σ
Stretch	97.97	8.07	10.94	6.84	99.48	7.68	8.27	3.47
Wave	99.12	6.90	8.07	4.00	100	6.16	6.15	1.54
Walk	99.40	3.11	6.50	4.11	98.96	3.20	7.22	5.39
Jog	98.27	5.97	8.22	4.29	97.52	4.11	7.98	5.87
Jump	87.82	16.62	34.76	47.36	43.12	53.40	254.74	278.75
Skip	92.51	20.19	29.50	14.45	75.19	24.30	80.69	116.43
Mean	95.85	10.14	16.33	13.51	83.71	16.47	60.84	68.57

Table 4
Global optimization vs local optimization.

Motion	MMTA (%)	MMTP (mm)	μ (mm)	σ (mm)
Stretch	78.18	36.39	101.65	27.29
Wave	81.20	32.22	86.65	40.87
Walk	78.55	43.27	91.68	25.11
Jog	70.23	47.12	100.14	30.88
Jump	62.31	49.52	150.77	53.10
Skip	74.78	49.75	102.59	46.61
Mean	74.21	43.04	105.58	37.31

Table 5
The result of the proposed algorithm with hard partitioning state space.

Motion	MMTA (%)	MMTP (mm)	μ (mm)	σ (mm)
Stretch	98.05	7.63	10.42	9.78
Wave	99.75	6.42	6.79	2.50
Walk	99.42	4.99	5.67	3.82
Jog	89.57	13.58	31.84	15.08
Jump	87.24	16.57	37.12	27.35
Skip	97.63	16.61	20.21	11.44
Mean	95.27	10.97	18.68	11.66

Table 6
Time consumption for each stage of the proposed algorithm.

Stage	Time for each frame (s)
Torso position and orientation	0.4
Left and right hand	0.7
Left and right leg	0.55
Head	0.3
Total	3.2

Table 7
Non-adaptive computation of particle velocity.

Motion	MMTA (%)	MMTP (mm)	μ (mm)	σ (mm)
Stretch	96.78	7.61	12.69	1.00
Wave	99.69	6.14	6.51	4.52
Walk	99.22	5.22	3.38	4.65
Jog	92.49	16.62	20.72	17.17
Jump	94.85	13.53	19.90	16.14
Skip	96.43	16.02	21.40	14.72
Mean	96.57	10.86	14.1	9.7

4.7. Swarm convergence

Visual representation of the stages of the proposed method is shown in Fig. 5. In this figure the swarm convergence is represented at stage four. As mentioned before, stages 1 to 6 are related to torso position and orientation, left and right hand, left and right leg and head orientation.

4.8. Computational time

The experiments have been tested on a 2.67 GHz Intel core2 CPU with 8 GB RAM, using MATLAB2013. Computation time for each stage is represented in Table 6.

4.9. Adaptive computation of particle's velocity

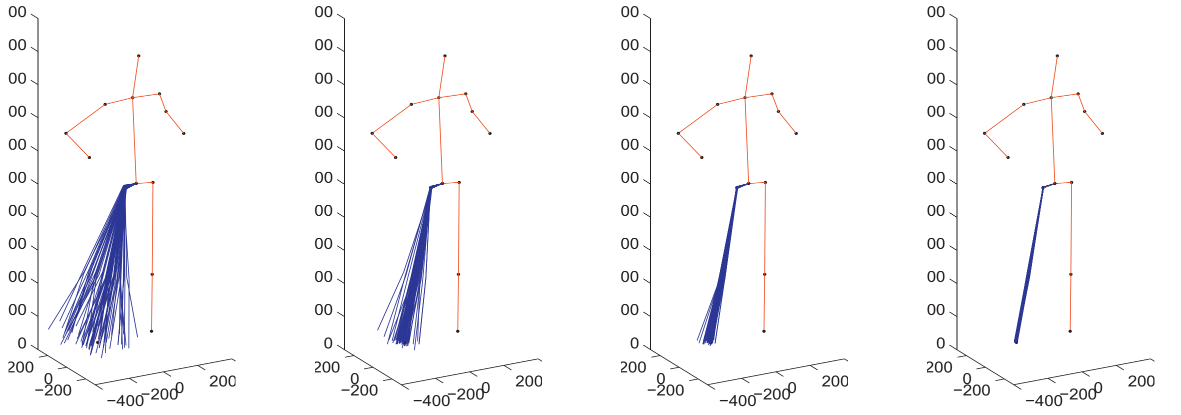
In this experiment we analyze the impact of adaptive computation of velocity of particles on the performance of the proposed algorithm. As mentioned in Section 3.4, proposed algorithm adaptively computes velocity of the particles based on variance of the parameters in each iteration. Adaptive computation of velocity prevents the dispersion of the particles after the convergence in specific iteration. The results of this experiment are represented in Table 7. Comparison of these results with result presented in Table 1 shows that in adaptive version of algorithm, estimated human pose converges better to correct human pose and performance of the algorithm has improved.

4.10. Comparison with PA and APF

In this section the proposed algorithm is compared with APF [17] and PA algorithm [34]. Experiments of APF were performed with 700 particles and 5 annealing layers and for PA algorithm with "vh" setup mentioned in [34]. Results of this experiment are represented in Table 8. As it can be seen, performance of the proposed algorithm is significantly better compared to APF and PA algorithms.

5. Conclusions

In this paper a combination of annealed PSO algorithm with search space partitioning is presented for marker-based pose estimation. The proposed algorithm uses a generic motion model that eliminated the need for training process. According to the experimental results, this algorithm is capable of tracking different motions with no prior knowledge about motion type. Utilizing a gradient-based method as a local refinement stage after optimization with the proposed algorithm, can be one area of future work which is expected to improve the efficiency of the pose estimation algorithm.



(a) Iteration=2 (b) Iteration=4 (c) Iteration=6 (d) Iteration=9

Fig. 5. Illustration of particle convergence in the proposed method.

Table 8
Comparison of mean tracking error (mm) of proposed tracking algorithm to APF and PA.

	MMTA (%)			MMTP		
Motion	Proposed method	PA	APF	Proposed method	PA	APF
Stretch	99.11	92.47	90.91	7.85	30.75	37.86
Wave	99.96	86.84	53.15	6.91	27.10	20.88
Walk	99.88	96.00	83.37	5.94	35.51	24.59
Jog	99.41	70.25	36.50	6.76	38.29	70.57
Jump	93.97	67.72	26.47	14.01	44.32	75.40
Skip	97.95	86.57	68.52	15.61	40.60	52.62
Mean	98.38	83.30	59.82	9.51	36.09	46.98
	μ (mm)			σ (mm)		
Motion	Proposed method	PA	APF	Proposed method	PA	APF
Stretch	8.97	56.63	69.08	4.30	25.80	30.33
Wave	6.95	75.85	81.94	1.91	30.49	38.54
Walk	6.05	42.68	79.67	3.04	11.20	22.55
Jog	6.84	115.50	247.56	2.04	41.05	75.36
Jump	21.14	138.45	311.05	16.54	59.42	138.36
Skip	18.07	69.11	118.01	6.27	27.87	46.70
Mean	11.33	83.03	151.21	5.68	32.63	58.64

References

[1] G.B. Guerra-Filho, Optical motion capture: theory and implementation, Journal of Theoretical and Applied Informatics (RITA 12 (2005) 61–89.

[2] T.B. Moeslund, E. Granum, A survey of computer vision-based human motion capture, Comput. Vis. Image Underst. 81 (3) (2001) 231–268. http://dx.doi.org/10.1006/cviu.2000.0897.

[3] T.B. Moeslund, F. Bajers, Computer Vision-based Human Motion Capture — A Survey, 1999.

[4] J.F. O'Brien, R.E. Bodenheimer, G.J. Brostow, J.K. Hodgins, Automatic joint parameter estimation from magnetic motion capture data, Proceedings of Graphics Interface 2000, 2000. pp. 53–60.

[5] P. Nogueira, Motion capture fundamentals, Faculdade de Engenharia da Universidade do Porto.

[6] C. Theobalt, M.A. Magnor, P. Schler, H.-P. Seidel, Combining 2D feature tracking and volume reconstruction for online video-based human motion capture, International Journal of Image and Graphics 04 (04) (2004) 563–583. http://dx.doi.org/10.1142/S0219467804001543.

[7] C. Canton-Ferrer, J.R. Casas, M. Pardàs, Marker-based Human Motion Capture in Multiview Sequences, EURASIP J. Adv. Signal Process 2010 (2010) 73:1–73:11. http://dx.doi.org/10.1155/2010/105476.

[8] T.B. Moeslund, A. Hilton, V. Krger, A survey of advances in vision-based human motion capture and analysis, Comput. Vis. Image Underst. 104 (23) (2006) 90–126. Special issue on modeling people: vision-based understanding of a persons shape, appearance, movement and behaviour. http://dx.doi.org/10.1016/j.cviu.2006.08.002.

[9] G. Pons-Moll, B. Rosenhahn, Visual Analysis of Humans: Looking at People, Ch. Model-based Pose Estimation, Springer London, London, 2011, 139–170.

[10] T.-J. Cham, J.M. Rehg, A multiple hypothesis approach to figure tracking, Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on., vol. 2, 1999. pp. 244 Vol. 2. http://dx.doi.org/10.1109/CVPR.1999.784636.

[11] R. Poppe, Vision-based human motion analysis: an overview, Comput. Vis. Image Underst. 108 (1-2) (2007) 4–18. http://dx.doi.org/10.1016/j.cviu.2006.10.016.

[12] J. Ohya, F. Kishino, Human posture estimation from multiple images using genetic algorithm, Pattern Recognition, 1994. Vol. 1 — Conference A: Computer Vision & Image Processing. Proceedings of the 12th IAPR International Conference on, vol. 1, 1994. pp. 750–753. vol. 1. http://dx.doi.org/10.1109/ICPR.1994.576430.

[13] X. Zhao, Y. Liu, Generative tracking of 3D human motion by hierarchical annealed genetic algorithm, Pattern Recogn. 41 (8) (2008) 2470–2483. http://dx.doi.org/10.1016/j.patcog.2008.01.004.

[14] M. Isard, A. Blake, CONDENSATION—conditional density propagation for visual tracking, Int. J. Comput. Vis. 29 (1) (2012) 5–28. http://dx.doi.org/10.1023/A:1008078328650.

[15] M. Behrouzifar, H. Shayegh Boroujeni, N. Moghadam Charkari, K. Mozafari, Knowledge Technology: Third Knowledge Technology Week, KTW 2011, Kajang, Malaysia, July 18–22, 2011. Revised Selected Papers, Ch. Model Based Human Pose Estimation in MultiCamera Using Weighted Particle Filters, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, 234–243. http://dx.doi.org/10.1007/978-3-642-32826-8_24.

[16] J. Deutscher, A. Blake, I. Reid, Articulated body motion capture by annealed particle filtering, Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on, vol. 2, 2000. pp. 126–133. vol. 2. http://dx.doi.org/10.1109/CVPR.2000.854758.

[17] J. Deutscher, I. Reid, Articulated body motion capture by stochastic search, Int. J. Comput. Vis. 61 (2) (2005) 185–205. http://dx.doi.org/10.1023/B:VISI.0000043757.18370.9c.

[18] L. Sigal, A.O. Balan, M.J. Black, HumanEva: synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion, Int. J. Comput. Vis. 87 (1) (2009) 4–27. http://dx.doi.org/10.1007/s11263-009-0273-6.

[19] J. Deutscher, A. Davison, I. Reid, Automatic partitioning of high dimensional search spaces associated with articulated body motion capture, Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, vol. 2, 2001. pp. II-669–II-676. vol. 2. http://dx.doi.org/10.1109/CVPR.2001.991028.

[20] N.H. Lehment, D. Arsi, M. Kaiser, G. Rigoll, Automated pose estimation in 3D point clouds applying annealing particle filters and inverse kinematics on a GPU, Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on, 2010. pp. 87–92. http://dx.doi.org/10.1109/CVPRW.2010.5543606.

[21] T. Krzeszowski, B. Kwolek, K. Wojciechowski, Articulated Body Motion Tracking by Combined Particle Swarm Optimization and Particle Filtering, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, 147–154. http://dx.doi.org/10.1007/978-3-642-15910-7_17. http://dx.doi.org/10.1007/978-3-642-15910-7_17.

[22] S. Ivekovic, E. Trucco, Human body pose estimation with PSO, Evolutionary Computation, 2006. CEC 2006. IEEE Congress on, 2006. pp. 1256–1263. http://dx.doi.org/10.1109/CEC.2006.1688453.

[23] J. MacCormick, M. Isard, Computer Vision — ECCV 2000: 6th European Conference on Computer Vision Dublin, Ireland, June 26–July 1, 2000 Proceedings, Part II, Springer Berlin Heidelberg, Berlin, Heidelberg, 2000, Ch. Partitioned Sampling, Articulated Objects, and Interface-quality Hand Tracking, 3–19, 10.1007/3-540-45053-X_1.

[24] P. Fleischmann, I. Austvoll, B. Kwolek, Advanced concepts for intelligent vision systems: 14th International Conference, ACIVS 2012, Brno, Czech Republic, September 4–7, 2012. Proceedings, Ch. Particle Swarm Optimization With Soft Search Space Partitioning for Video-based Markerless Pose Tracking, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, 479–490. http://dx.doi.org/10.1007/978-3-642-33140-4_42.

[25] J. Bandouch, F. Engstler, M. Beetz, Evaluation of hierarchical sampling strategies in 3D human pose estimation, Proceedings of the 19th British Machine Vision Conference (BMVC), 2008.

[26] A. Sharifi, A. Harati, A. Vahedian, Marker based human pose estimation using annealed particle swarm optimization with search space partitioning, Computer and Knowledge Engineering (ICCKE), 2014 4th International eConference on, 2014. pp. 135–140. http://dx.doi.org/10.1109/ICCKE.2014.6993366.

[27] M.M.S. Maddock, Motion Capture File Formats Explained.

[28] S. Ivekovic, V. John, E. Trucco, Applications of Evolutionary Computation: EvoApplicatons 2010: EvoCOMPLEX, EvoGAMES, EvoIASP, EvoINTELLIGENCE, EvoNUM, and EvoSTOC, Istanbul, Turkey, April 7–9, 2010, Proceedings, Part I, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010 Ch. Markerless Multi-view Articulated Pose Estimation Using Adaptive Hierarchical Particle Swarm Optimisation, 241–250, 10.1007/978-3-642-12239-2_25.

[29] A. Malian, A. Azizi, F.A. van den Heuvel, M. Zolfaghari, Development of a robust photogrammetric metrology system for monitoring the healing of bedsores, Photogramm. Rec. 20 (111) (2005) 241–273. http://dx.doi.org/10.1111/j.1477-9730.2005.00319.x. http://dx.doi.org/10.1111/j.1477-9730.2005.00319.x.

[30] Y. del Valle, G.K. Venayagamoorthy, S. Mohagheghi, J.C. Hernandez, R.G. Harley, Particle swarm optimization: basic concepts, variants and applications in power systems, IEEE Trans. Evol. Comput. 12 (2) (2008) 171–195. http://dx.doi.org/10.1109/TEVC.2007.896686.

[31] X. Wang, X. Zou, W. Wan, X. Yu, Articulated 3D human pose estimation with particle filter based particle swarm optimization, Audio Language and Image Processing (ICALIP), 2010 International Conference on, 2010. pp. 1094–1099. http://dx.doi.org/10.1109/ICALIP.2010.5685102.

[32] X. Zhao, Y. Liu, PEAR: Synchronized 16-views visual and 3D human pose dataset for pose estimation and action recognition, Tech. Rep., Shanghai Jiao Tong University, Shanghai, China, 2012.

[33] C. Canton-Ferrer, J. Casas, M. Pardas, E. Monte, Towards a Fair Evaluation of 3D Human Pose Estimation Algorithms, Technical University of Catalonia, Barcelona, Spain, 2009.

[34] P. Kaliamoorthi, R. Kakarala, Parametric annealing: a stochastic search method for human pose tracking, Pattern Recogn. 46 (5) (2013) 1501–1510. http://dx.doi.org/10.1016/j.patcog.2012.11.005.